Major Shared Resource Center

ERDC MSRC

ERDC MSRC/PET TR/00-24

Modification of the External Mode Solver in CH3D-Z

by

Clint Dawson
Dharhas Potina
Mary F. Wheeler

16 June 2000

Modification of the External Mode Solver in CH3D-Z

Clint Dawson
Dharhas Potina
and
Mary F. Wheeler
Center for Subsurface Modeling
University of Texas, Austin

June 16, 2000

1 Introduction

CH3D-Z is a well-known hydrodynamic, salinity and temperature model developed at the U.S. Army Engineer Research and Development Center (ERDC), Vicksburg MS [1]. In CH3D-Z, finite differences are used to replace derivatives in the governing equations, resulting in a system of linear algebraic equations. The code solves external mode equations for depth-averaged velocities and surface elevations, and internal mode equations to account for three-dimensional variations in velocities, temperature and salinity.

The external mode consists of equations for the water surface elevation ζ and vertically integrated contravariant unit flows $\overline{U}$ and $\overline{V}$. All of the terms in the transformed vertically averaged continuity equations are treated implicitly, whereas only the water surface slope terms in the transformed vertically averaged momentum equations are treated implicitly. Those terms treated implicitly are weighted between the new and old time steps. In the original solver the resulting finite difference equations were factored such that a ξ-sweep followed by an η-sweep of the horizontal grid yielded the solution at the new time step. We have modified the code so that the equations for water surface elevation and vertically integrated contravariant unit flows have been combined into a single equation in terms of ζ, which is then solved to yield the solution at the new time step. This global, linear system in ζ can be solved by standard iterative techniques. We have employed a preconditioned conjugate gradient algorithm to this system. By replacing the directional sweeps with a global solve involving a sparse, positive definite matrix, we have obtained an algorithm which can be easily parallelized for massively parallel computers. This parallization will be the subject of future work.

The external mode equations in computational space are given by:

$$\begin{aligned}
\frac{\partial\zeta}{\partial t} + \beta(\frac{\partial\overline{U}}{\partial\xi} + \frac{\partial\overline{V}}{\partial\eta}) &= 0, && (1)\\
\frac{\partial\overline{U}}{\partial t} + \frac{H}{J^2}G_{22}\frac{\partial\zeta}{\partial\xi} &= M, && (2)\\
\frac{\partial\overline{V}}{\partial t} + \frac{H}{J^2}G_{11}\frac{\partial\zeta}{\partial\eta} &= N, && (3)
\end{aligned}$$

where M and N include advective and forcing terms.

2 Implementation

The code uses a staggered grid approach, where elevations are approximated at the cell centers, and velocities are approximated at the cell edges. Let i, j subscripts denote the value of ζ at the center of grid block i, j, the value of $\overline{U}$ on the left edge of the grid block, and the value of $\overline{V}$ on the bottom edge of the grid block.

In the original solver the values at the new time step were computed by using the following equations in a ξ-sweep and an η-sweep.

The ξ - sweep :

$$\begin{aligned}
\zeta^*_{ij} + \frac{\beta\theta\Delta t}{\Delta\xi}(\overline{U}^*_{i+1,j} - \overline{U}^*_{i,j}) &= \zeta^n_{i,j}\\
&\quad -(1-\theta)\frac{\Delta t}{\Delta\xi}(\overline{U}^n_{i+1,j} - \overline{U}^n_{i,j})\\
&\quad -\frac{\Delta t}{\Delta\eta}(\overline{V}^n_{i,j+1} - \overline{V}^n_{i,j}). && (4)
\end{aligned}$$

$$\begin{aligned}\overline{U}^{n+1}_{i,j} + \frac{\theta\Delta t H G_{22}}{\Delta\xi J^2}(\zeta^*_{i,j} - \zeta^*_{i-1,j}) &= \overline{U}^{n}_{i,j} \\ &\quad -(1-\theta)\frac{\theta\Delta t H G_{22}}{\Delta\xi J^2}(\zeta^n_{i,j} - \zeta^n_{i-1,j}) \\ &\quad +\Delta t M^n. \end{aligned} \tag{5}$$

The η - sweep :

$$\begin{aligned}\zeta^{n+1}_{ij} + \frac{\beta\theta\Delta t}{\Delta\eta}(\overline{V}^{n+1}_{i,j+1} - \overline{V}^{n+1}_{i,j}) &= \zeta^*_{i,j} \\ &\quad -(1-\theta)\frac{\Delta t}{\Delta\eta}(\overline{V}^{n}_{i,j+1} - \overline{V}^{n}_{i,j}) \\ &\quad -\frac{\Delta t}{\Delta\eta}(\overline{V}^{n}_{i,j+1} - \overline{V}^{n}_{i,j}). \end{aligned} \tag{6}$$

$$\begin{aligned}\overline{V}^{n+1}_{i,j} + \frac{\theta\Delta t H G_{11}}{\Delta\eta J^2}(\zeta^{n+1}_{i,j} - \zeta^{n+1}_{i,j-1}) &= \overline{V}^{n}_{i,j} \\ &\quad -(1-\theta)\frac{\theta\Delta t H G_{11}}{\Delta\eta J^2}(\zeta^n_{i,j+1} - \zeta^n_{i,j}) \\ &\quad +\Delta t N^n. \end{aligned} \tag{7}$$

Here we have suppressed the spatial dependence of the coefficients G_{11}, G_{22} and J, which involve the mapping from the physical domain to the computational grid.

We now combine (4), (5), (6) and (7) to get the single equation in ζ :

$$\begin{aligned} &2[1 + (\frac{\beta\theta\Delta t}{\Delta\xi})(\frac{\theta\Delta t H G_{22}}{\Delta\xi J^2}) + \\ &\quad (\frac{\beta\theta\Delta t}{\Delta\eta})(\frac{\theta\Delta t H G_{11}}{\Delta\eta J^2})]\zeta^{n+1}_{i,j} \\ &-[(\frac{\beta\theta\Delta t}{\Delta\xi})(\frac{\theta\Delta t H G_{22}}{\Delta\xi J^2})]\zeta^{n+1}_{i-1,j} \end{aligned}$$

$$\begin{aligned}
-[(\frac{\beta\theta\Delta t}{\Delta\xi})(\frac{\theta\Delta t H G_{22}}{\Delta\xi J^2})]\zeta^{n+1}_{i+1,j} & \\
-[(\frac{\beta\theta\Delta t}{\Delta\eta})(\frac{\theta\Delta t H G_{11}}{\Delta\eta J^2})]\zeta^{n+1}_{i,j-1} & \\
-[(\frac{\beta\theta\Delta t}{\Delta\eta})(\frac{\theta\Delta t H G_{11}}{\Delta\eta J^2})]\zeta^{n+1}_{i,j+1} &= R1_{i,j} \\
&\quad -\frac{\beta\theta\Delta t}{\Delta\xi}R2_{i+1,j}+\frac{\beta\theta\Delta t}{\Delta\xi}R2_{i,j} \\
&\quad -\frac{\beta\theta\Delta t}{\Delta\eta}R3_{i,j+1}+\frac{\beta\theta\Delta t}{\Delta\eta}R3_{i,j} \qquad (8)
\end{aligned}$$

where :

$$\begin{aligned}
R1_{i,j} &= \zeta^n_{i,j}-(1-\theta)\frac{\Delta t}{\Delta\xi}(\overline{U}^n_{i+1,j}-(\overline{U}^n_{i,j})-\frac{\Delta t}{\Delta\eta}(\overline{V}^n_{i,j+1}-(\overline{V}^n_{i,j}) \\
&\quad +\zeta^n_{i,j}-(1-\theta)\frac{\Delta t}{\Delta\eta}(\overline{V}^n_{i,j+1}-(\overline{V}^n_{i,j})+\frac{\Delta t}{\Delta\eta}(\overline{V}^n_{i,j+1}-(\overline{V}^n_{i,j}) \qquad (9)
\end{aligned}$$

$$R2_{i,j} = \overline{U}^n_{i,j}-(1-\theta)\frac{\Delta t H G_{22}}{\Delta\xi J^2}(\zeta^n_{i,j}-\zeta^n_{i-1,j})+\Delta t M^n \qquad (10)$$

$$R3_{i,j} = \overline{V}^n_{i,j}-(1-\theta)\frac{\Delta t H G_{11}}{\Delta\eta J^2}(\zeta^n_{i,j+1}-\zeta^n_{i,j})+\Delta t N^n \qquad (11)$$

Using this equation we can assemble a matrix which is then solved using the NSPCG package to yield ζ at the new time step. $\overline{U}$ and $\overline{V}$ at the new time step are then retrieved using (5) and (7).

3 Coding Details

The subroutine that required modification was CH2DXY. The matrix was constructed entirely by using the coefficients already calculated in the original code.

The coefficient matrix is stored in the array COEF(K,L), where the K index ranges from 1 to the number of unknowns, and L ranges from 1 to 5. The unknowns $\zeta_{i,j}$ are ordered with j increasing most rapidly. The indices L have the following correspondence:

L	corresponding unknown
1	i, j
2	$i-1, j$
3	$i, j-1$
4	$i, j+1$
5	$i+1, j$

The right hand side is stored in the vector RHS.

4 Calling NSPCG

Once COEF and RHS are constructed, the package NSPCG is used to solve the linear system. In particular, we use the diagonally preconditioned conjugate gradient routine within the NSPCG package. Complete details on the NSPCG package can be found in the user's manual, which is available at the web site

`http://rene.ma.utexas.edu/CNA/NSPCG/`

We outline just a few details needed to implement the method within CH2DXY.

First, one must call the NPSCG subroutine DFAULT to set default values for certain solver parameters. After calling DFAULT, these parameters can be changed if desired. We have reset IPARM(1) = 10, which controls the stopping test for the solver, and we have reset the solver tolerance RPARM(1) to .00001. IPARM(12) has been reset to 1, which indicates to the solver the storage format used for storing the matrix coefficients. Workspace parameters INW and NW are then computed based on formulas given in the user's guide. Since we are using Jacobi preconditioned conjugate gradient, the preconditioner routine JAC1 and the accelerator CG must be declared as EXTERNAL variables in the CH2DXY. Real and integer workspace variables WKSP and IWKSP must also be declared, with sufficiently large memory. A real variable UBAR and integer variables P and IP must also be declared, but these are not used in the solver. The call to NSPCG is then

```
  CALL NSPCG(JAC1, CG, IC1*JC1, 5, IC1*JC1, 5, COEF, JCOEF, P,
$      IP, SN1, UBAR, RHS, WKSP, IWKSP, NW, INW, IPARM, RPARM, IER)
```

5 Test runs

The code has been tested on a Chesapeake Bay data set provided by ERDC. The simulated time in these runs was 30 days. Comparison of surface elevations for the new code and the original code can be found in Figures 1 and 1, a plot of the differences in surface elevations between the two codes can be found in Figure 3. Note that we expect some changes in the output because we are using a different time-stepping scheme in the new code. Plots of the differences in the u and v components of velocity can be found in Figures 4 and 5.

Figure 1 Plot of surface elevation using new solver

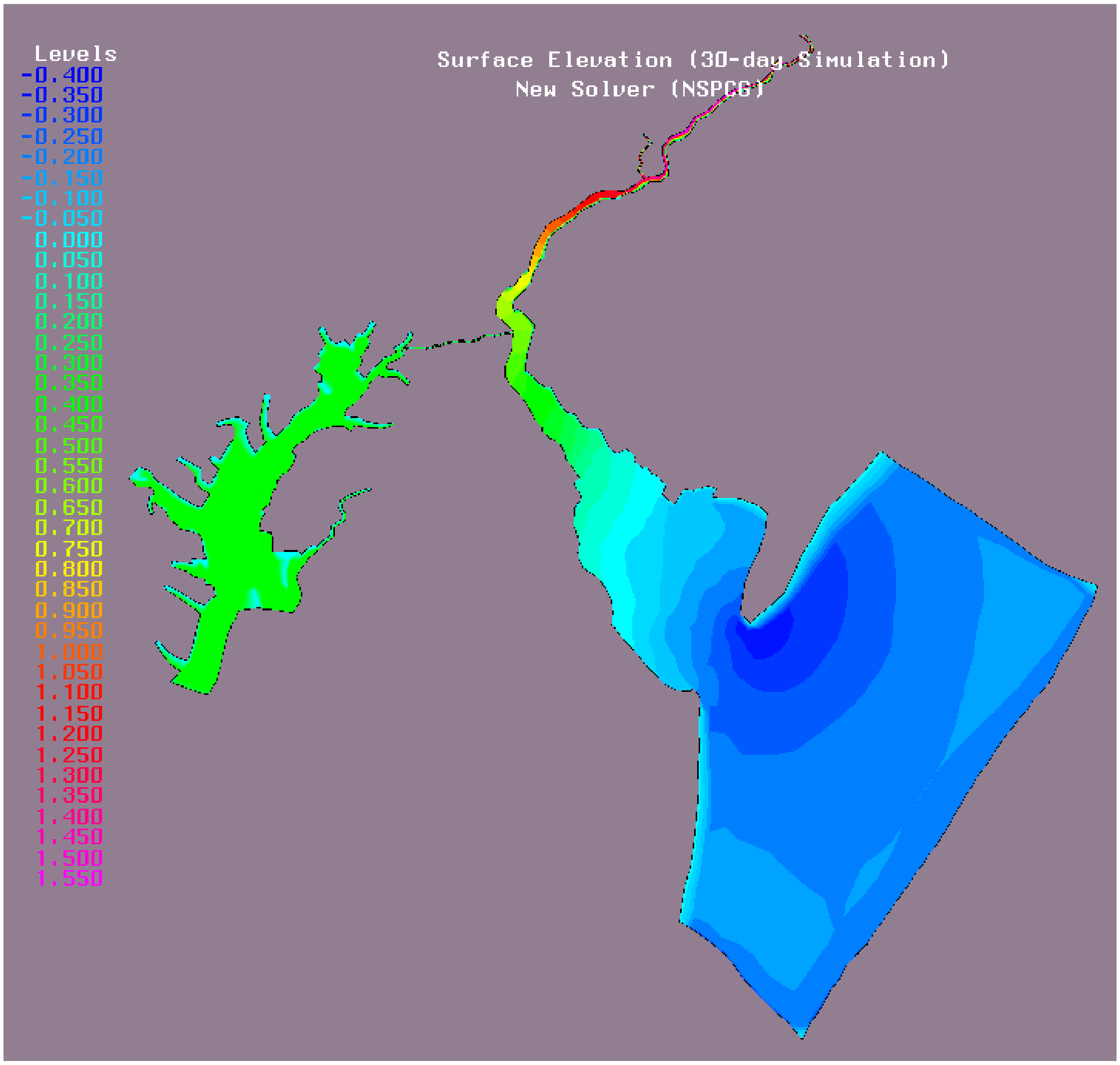


Figure 2 Plot of surface elevation using original solver

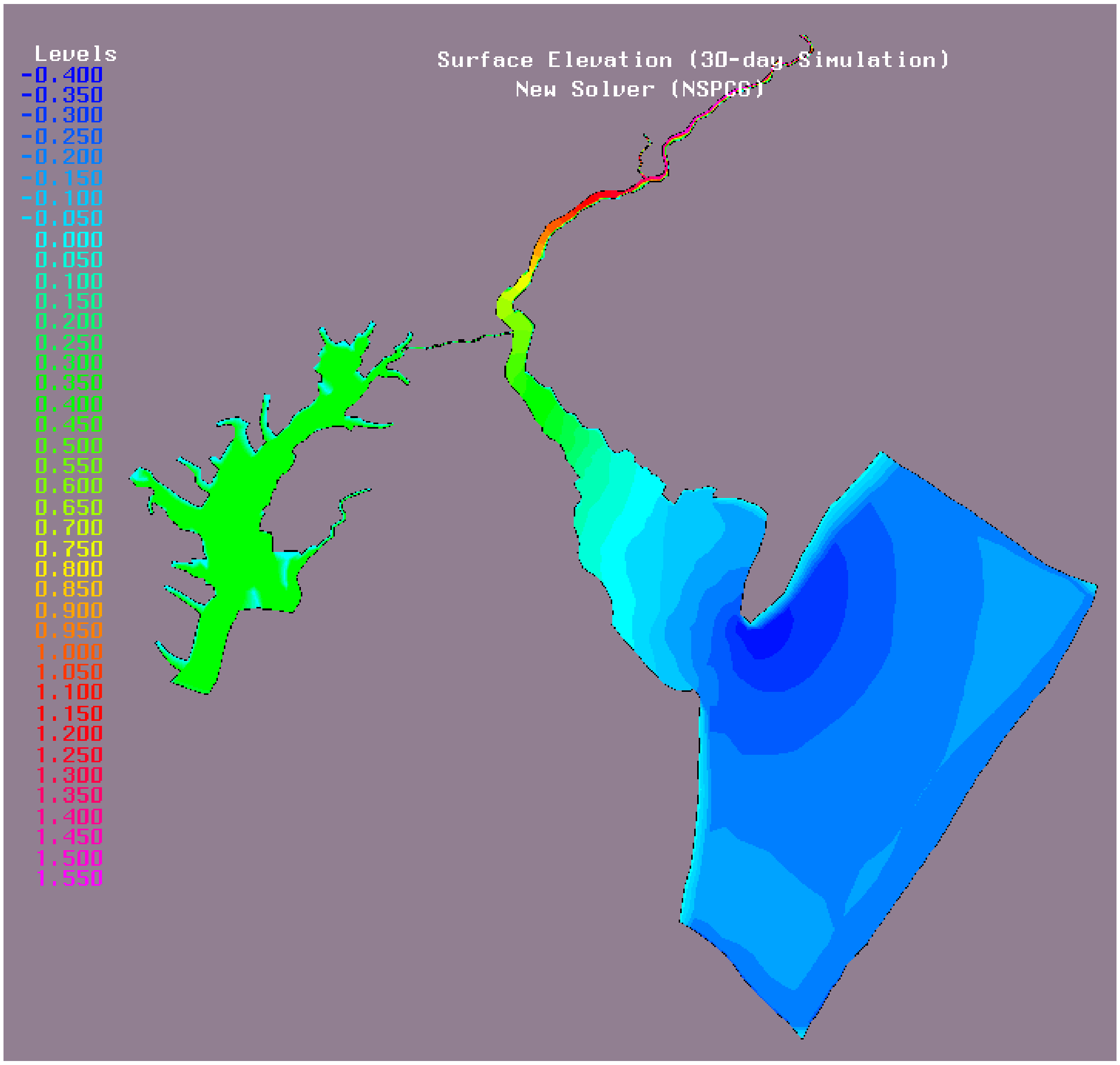


Figure 3 Differences in surface elevation between the new and original solvers

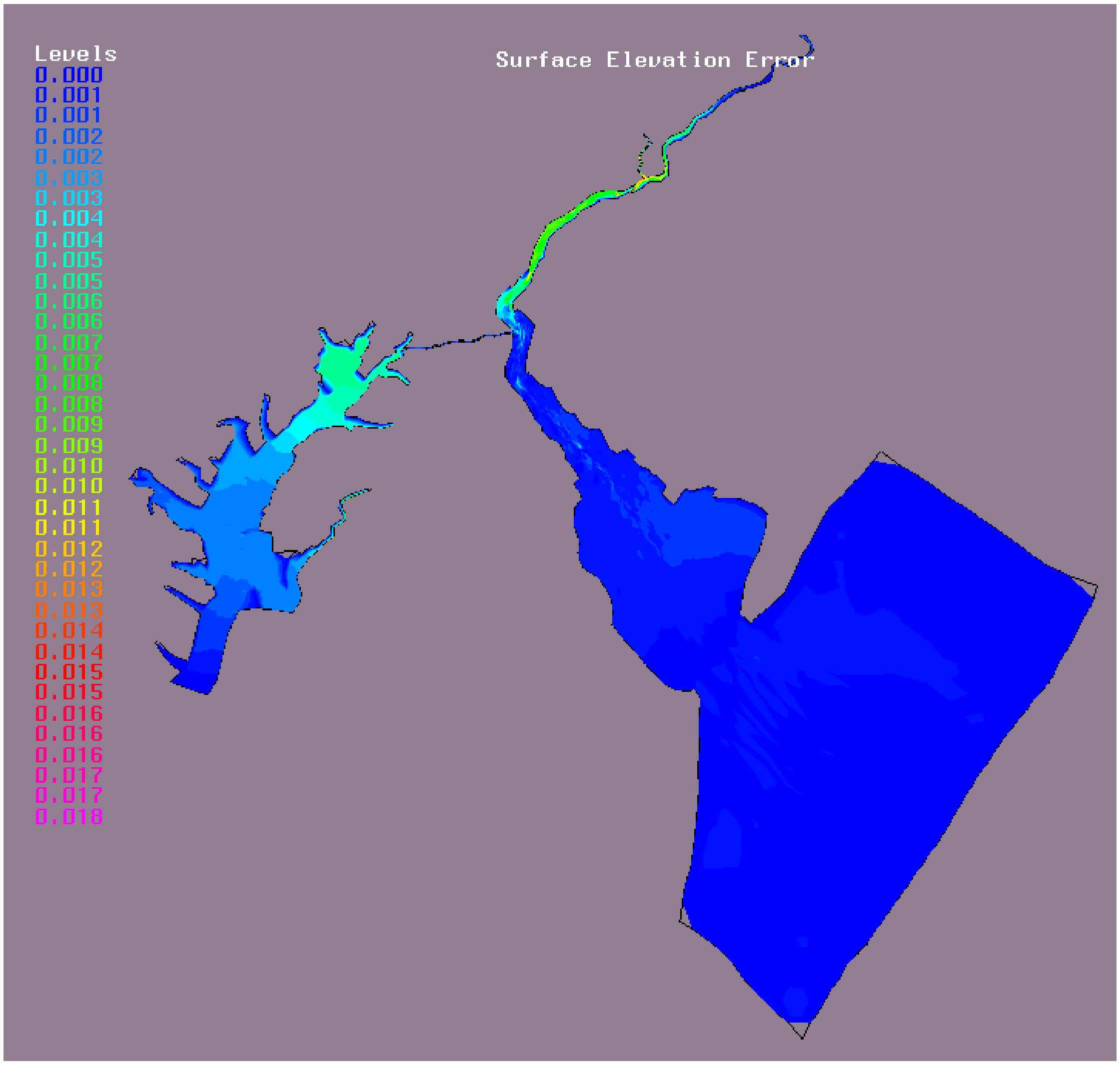


Figure 4 Differences in u component between the new and original solvers

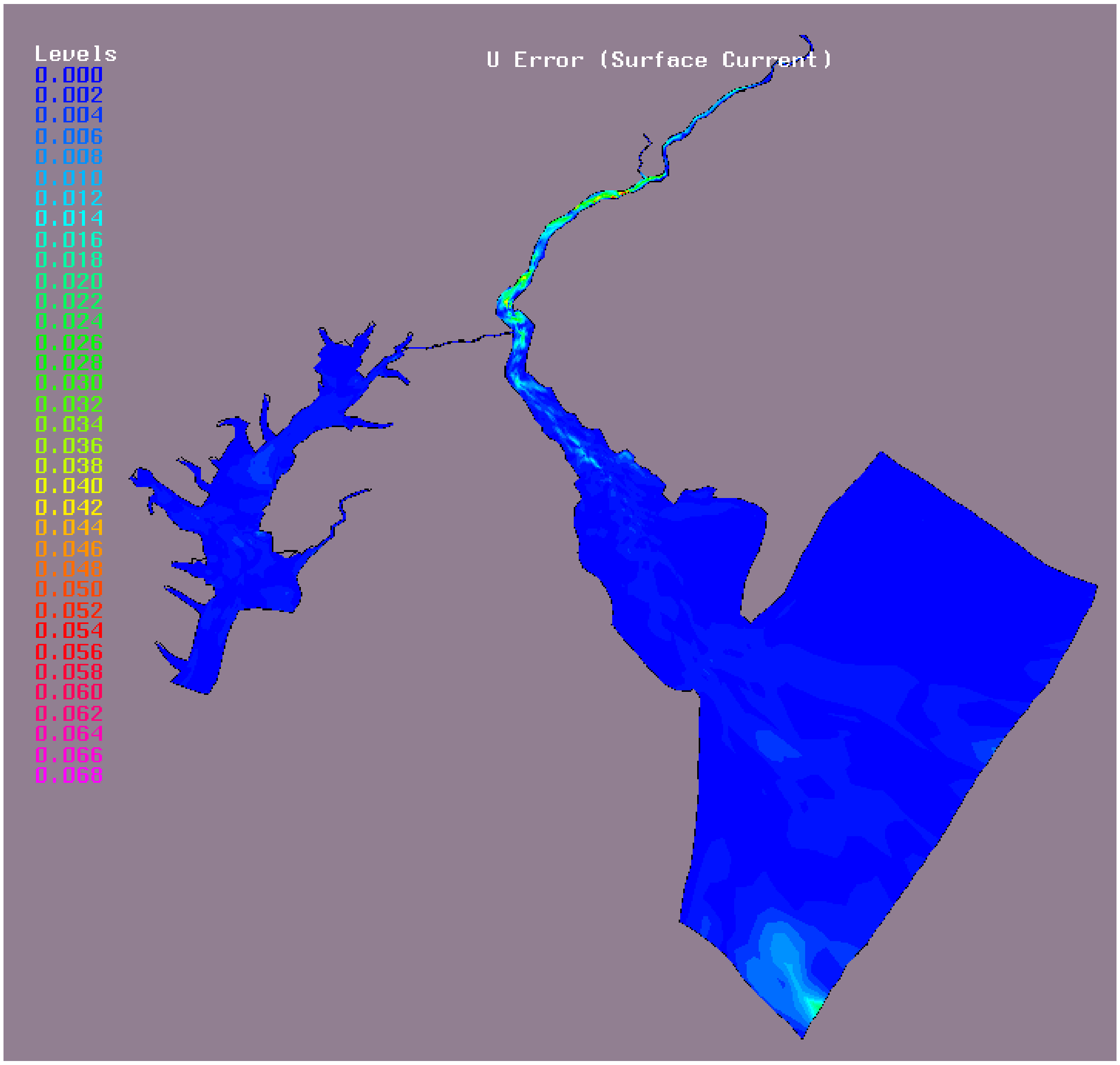
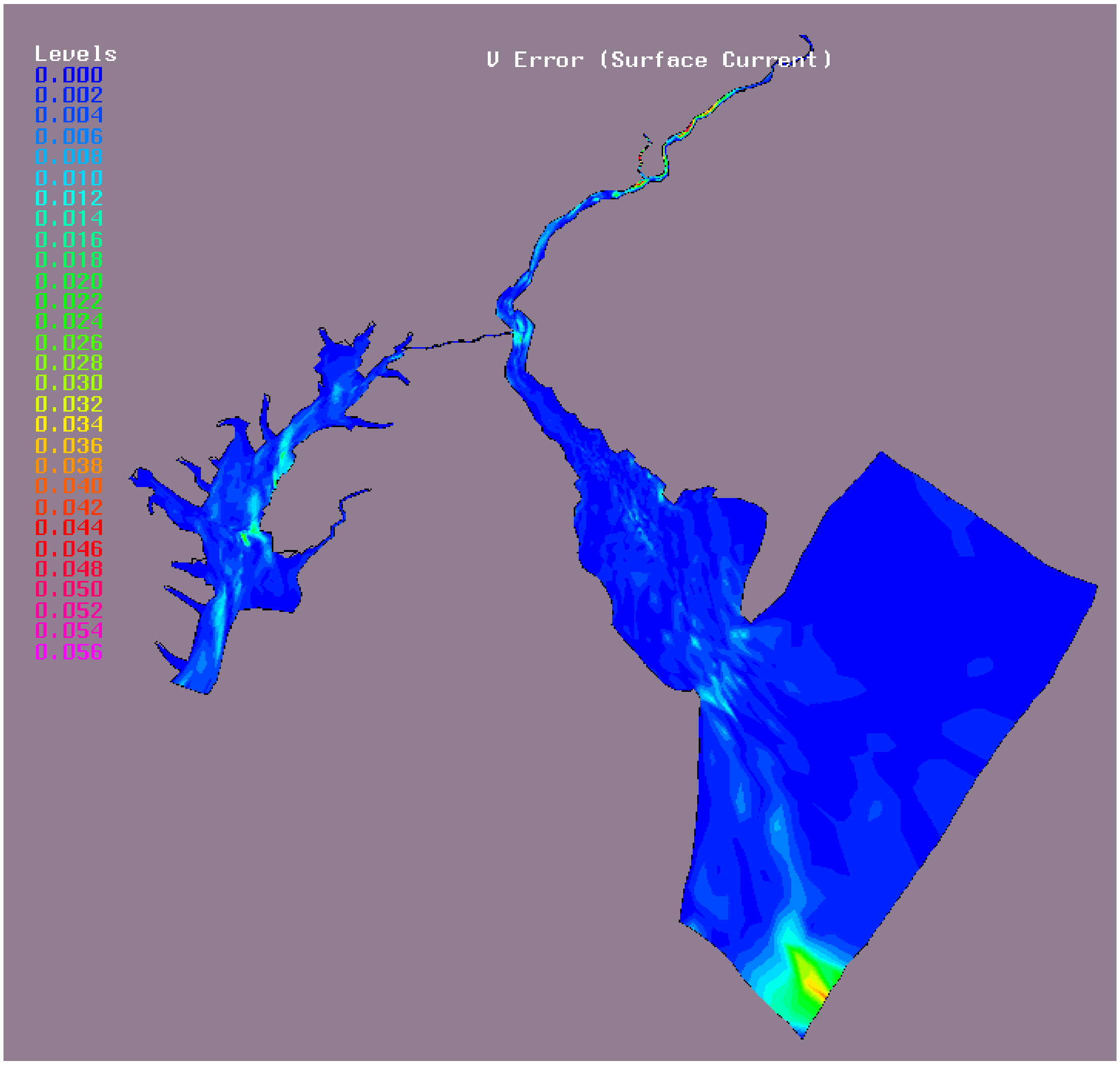


Figure 5 Differences in v component between the new and original solvers



6 Conclusions

Currently, the new solver requires about 10% more CPU time to execute than the original solver on the Chesapeake Bay data set. This is primarily due to the fact that we are including keyed-out elements in the matrix, of which there are many in the Chesapeake Bay grid. This can be improved by removing these unknowns from the matrix.

One of the advantages of this new iterative approach is its inherent parallelizability. We plan to take advantage of this fact in the development of a new parallel version of CH3D-Z in the coming year.

References

[1] B.H. Johnson, R.E. Heath, B.B. Hsieh, K.W. Kim and H.L. Butler, *Development and verification of a three-dimensional numerical hydrodynamic, salinity, and temperature model of Chesapeake Bay, Volume 1*, Technical Report HL-91-7, Department of the Army, Waterways Experiment Station, Corps of Engineers, 3909 Halls Ferry Road, Vicksburg, MS, 39180-6199, 1991.